

## META-HEURÍSTICAS BIO-INSPIRADAS APLICADAS AO PROBLEMA DE FLUXO DE POTÊNCIA ÓTIMO REATIVO

**Bruno Rafael Gamino**

Universidade Estadual Paulista  
Av. Brasil 56, Centro, 15385-000, Ilha Solteira, SP, Brasil  
gaminobr@gmail.com

**Percival Bueno de Araujo**

Universidade Estadual Paulista  
Av. Brasil 56, Centro, 15385-000, Ilha Solteira, SP, Brasil  
percival@dee.feis.unesp.br

**Elenilson de Vargas Fortes**

Instituto Federal de Goiás  
Rua Maria Vieira Cunha 775, Residencial Flamboyant, 75804-714, Jataí, GO, Brasil  
elenilson.fortes@ifg.edu.br

**Luís Fabiano Barone Martins**

Instituto Federal do Paraná  
Av. Doutor Tito s/n, Jardim Panorama, 86400-000, Jacarezinho, PR, Brasil  
luis.martins@ifpr.edu.br

### RESUMO

Neste artigo são aplicadas três meta-heurísticas bio-inspiradas (Otimização por Enxame de Partículas, Algoritmo Genético e Algoritmo Genético de Chu e Beasley) no ajuste das variáveis de controle do modelo de fluxo de potência ótimo reativo, cujo objetivo é minimizar as perdas de potência ativa de um sistema de energia elétrica, atendendo às suas restrições físicas e operacionais. Um estudo comparativo é realizado entre as soluções fornecidas pelos métodos de otimização a partir de simulações nos sistemas IEEE 57, 118 e 300. Os resultados obtidos indicam que o Algoritmo Genético de Chu e Beasley apresenta um desempenho superior para o tipo de problema proposto.

**PALAVRAS CHAVE.** Algoritmos Genéticos. Fluxo de Potência Ótimo. Otimização por Enxame de Partículas.

**Tópico:** EN - PO na Área de Energia

### ABSTRACT

In this article are applied three bio-inspired metaheuristics (Particle Swarm Optimization, Genetic Algorithm and Genetic Algorithm of Chu-Beasley's) in the setting of the control variables of the reactive optimal power flow model whose objective is minimize the active power losses of an electric energy system, taking into account their physical and operational constraints. A comparative study is made between the solutions provided by optimization methods from simulations in the systems IEEE 57, 118 and 300. The results indicate that the Genetic Algorithm of Chu-Beasley's shows superior performance for the type of the problem proposed.

**KEYWORDS.** Genetic Algorithms. Optimal Power Flow. Particle Swarm Optimization.

**Paper Topic:** EN - OR Area of Energy

## 1. Introdução

Em um cenário com mercados energéticos cada vez mais competitivos e descentralizados, diversos programas computacionais são utilizados no planejamento e operação dos Sistemas de Energia Elétrica (SEE). A intenção é garantir a qualidade e a confiabilidade dos SEEs gerando os menores custos possíveis e garantindo a maximização dos lucros.

Dentre estas ferramentas computacionais, uma das mais importantes se trata do Fluxo de Potência Ótimo (FPO). A necessidade da análise do problema de FPO tem como origem a topologia da rede, haja vista o fato das cargas e as unidades geradoras não se encontrarem conectadas à mesma barra. Em vista disto, torna-se possível o aparecimento de grandezas (tensão, fluxo de potência, etc) na rede elétrica com níveis impraticáveis, provocando funcionamento inadequado com perdas ativas elevadas e prejuízos desnecessários.

O problema de FPO foi proposto no início da década de 60, a partir do problema de despacho econômico [Carpentier, 1962]. Desde então, tem sido amplamente utilizado em SEEs como uma ferramenta computacional que visa ajustar as variáveis de controle a fim de obter um estado da rede elétrica que otimiza uma função objetivo, como por exemplo, o custo total de geração ou a minimização das perdas ativas, levando em consideração um conjunto de restrições físicas e operacionais [Zhu, 2009].

Geralmente, devido à sua complexidade, o problema de FPO é desacoplado e dividido em dois subproblemas: ativo e reativo. Dessa forma, cada subproblema considera as restrições e variáveis de controle de interesse [Papalexopoulos et al., 1989]. Neste trabalho é utilizado um modelo reativo (FPOR), cujo objetivo é minimizar as perdas ativas de um sistema elétrico a partir do chaveamento de bancos de capacitores e reatores *shunt*, além do ajuste da tensão nas unidades geradoras e a posição dos taps dos transformadores em fase.

Sendo caracterizado como um problema de programação não-linear inteiro mista, não convexo e de grande porte, várias técnicas de solução exata de modelos aproximados ou relaxados foram utilizadas para o FPO no decorrer dos anos, dentre elas, podemos citar as técnicas que utilizam métodos de gradientes [Dommel e Tinney, 1968] e de pontos interiores [Quintana et al., 2000; Jabr et al., 2002; Ramos et al., 2005; Sousa et al., 2011].

Com o passar dos anos, métodos estocásticos como as meta-heurísticas ganharam destaque na resolução do problema de FPO. Nestes métodos, a busca de uma solução é realizada através de regras probabilísticas, sem a necessidade de modelos de alta complexidade. Além disso, as meta-heurísticas, quando ajustadas convenientemente, possuem facilidade para sair de ótimos locais e trabalhar com variáveis contínuas, discretas e binárias simultaneamente [Bakirtzis et al., 2002; Biskas et al., 2005; Bouchekara et al., 2014].

Dentre as meta-heurísticas mais aplicadas em problemas de engenharia, encontra-se o Algoritmo Genético (AG). Baseado na biologia evolutiva, ele incorpora elementos como hereditariedade, seleção natural, recombinação e mutação [Holland, 1992]. Trata-se de um método flexível com alta capacidade de produção de soluções de boa qualidade em problemas complexos e de grande porte [Glover e Kochenberger, 2003].

Uma versão modificada do AG foi proposta em 1997 e ficou conhecida como Algoritmo Genético de Chu and Beasley (AGCB). Entre as mudanças propostas estavam a adição de uma etapa de melhoria local, a avaliação das factibilidades e infactibilidades de forma separada e um novo conceito de substituição dos indivíduos na população, garantindo a permanência das soluções de excelente qualidade [Chu e Beasley, 1997].

Baseado no comportamento de um grupo de pássaros em busca de alimentos, um algoritmo denominado *Particle Swarm Optimization* (PSO) foi concebido por Kennedy e Eberhart [1995]. Eles observaram que cada indivíduo realizava mudanças específicas na sua posição, sempre observando os movimentos dos demais indivíduos e respeitando conceitos como harmonia, separação e alinhamento do grupo. Desde então, esta técnica tem recebido diversas aplicações, inclusive ao problema de FPO [Abido, 2002].

Diante do exposto, as principais contribuições deste trabalho são: 1) Utilizar três meta-heurísticas (AG, AGCB e PSO) no ajuste das variáveis de controle de um modelo FPOR, objetivando minimizar as perdas ativas dos sistemas testes IEEE 57, 118 e 300; 2) Realizar um estudo comparativo entre os métodos utilizados a fim de concluir sobre a técnica melhor adaptada a este tipo de problema.

## 2. O Modelo de Fluxo de Potência Ótimo Reativo

### 2.1. Formulação Matemática

As tensões nas barras dos geradores, os taps dos transformadores em fase e as susceptâncias equivalentes dos bancos de capacitores e reatores *shunt* são as variáveis de controle do modelo, as quais são tratadas neste trabalho, como variáveis contínuas, discretas e binárias, respectivamente.

O modelo matemático do FPOR utilizado é expresso nas equações (1a)–(1h). A equação (1a) representa as perdas ativas que devem ser minimizadas. As equações (1b) e (1c) são as restrições de igualdade do modelo e estão relacionadas ao balanço de potência. As restrições de desigualdade dizem respeito ao conjunto de restrições funcionais e de operação e englobam os limites: de geração de potência reativa nas barras de geração (1d), do fluxo de potência nas linhas (1e) e das magnitudes das tensões nas barras (1f). Os limites das variáveis de controle também são apresentadas nas equações (1g) e (1h).

$$\min \sum_{k,m \in \mathcal{L} \cup \mathcal{A}} g_{km} \left( \frac{1}{a_{km}^2} V_k^2 + V_m^2 - \frac{2}{a_{km}} V_k V_m \cos \theta_{km} \right) \quad (1a)$$

$$\text{s.a:} \quad P_k - \sum_{m \in \Omega_k} P_{km} = 0 \quad \forall k \in \mathcal{B}' \quad (1b)$$

$$Q_k + Q_k^{sh} - \sum_{m \in \Omega_k} Q_{km} = 0 \quad \forall k \in \mathcal{B}_c \quad (1c)$$

$$Q_{gk}^{min} \leq Q_{gk} \leq Q_{gk}^{max} \quad \forall k \in \mathcal{B}_g \quad (1d)$$

$$-S_{km}^{max} \leq S_{km} \leq S_{km}^{max} \quad \forall k, m \in \mathcal{L} \quad (1e)$$

$$V_k^{min} \leq V_k \leq V_k^{max} \quad \forall k \in \mathcal{B} \quad (1f)$$

$$a_{km} \in \mathcal{D}_{km}^{tap} \quad \forall k, m \in \mathcal{A} \quad (1g)$$

$$b_k^{sh} \in \{0, 1\} \quad \forall k \in \mathcal{B}^{sh} \quad (1h)$$

Nas equações (1a)–(1h), os conjuntos  $\mathcal{B}_c$  e  $\mathcal{B}_g$  representam, nesta sequência, as barras de carga e de geração do sistema.  $\mathcal{B}$  é o conjunto de todas as barras do sistema e  $\mathcal{B}'$  é uma variação de  $\mathcal{B}$ , sendo obtido com a exclusão da barra *slack* de referência.  $\mathcal{B}^{sh}$  é o conjunto das barras com possibilidade de controle da magnitude das tensões, pois estão equipadas com os bancos de capacitores e reatores *shunt*.  $\mathcal{L}$  representa o conjunto das linhas de transmissão.  $\Omega_k$  é o conjunto de barras vizinhas à barra  $k$ .  $\mathcal{A}$  é o conjunto dos ramos  $k - m$  que representam os transformadores em fase com tap variável e  $\mathcal{D}_{km}^{tap}$  é um conjunto discreto com os valores que os taps podem assumir.

Com relação às variáveis e grandezas do modelo têm-se que  $V_k$  e  $\theta_k$  representam o módulo e a fase da tensão em uma barra  $k$ , e ainda,  $\theta_{km}$  é a diferença angular das tensões nas barras  $k$  e  $m$ . Os taps variáveis dos transformadores em fase e a condutância série em uma linha de transmissão que interliga as barras  $k$  e  $m$  são representados por  $a_{km}$  e  $g_{km}$ , respectivamente.  $P_k$  e  $Q_k$  são, nesta ordem, as potências líquidas ativa e reativa injetadas na barra  $k$ .  $P_{km}$ ,  $Q_{km}$  e  $S_{km}$  representam os

fluxos de potência ativa, reativa e aparente na linha de transmissão que conecta as barras  $k$  e  $m$ .  $Q_{gk}$  é a potência reativa gerada na barra  $k$ , onde seu limite inferior e superior é indicado por  $Q_{gk}^{min}$  e  $Q_{gk}^{max}$ . A susceptância equivalente do banco de capacitores e reatores *shunt* e a potência reativa injetada pelo banco em uma barra  $k$  são representados, nesta ordem, por  $b_k^{sh}$  e  $Q_k^{sh}$ . Por fim, têm-se que  $V_k^{min}$ ,  $V_k^{max}$  e  $S_{km}^{max}$  são os limites inferiores e superiores para a tensão na barra  $k$  e o limite superior para o fluxo de potência aparente entre as barras  $k$  e  $m$ , respectivamente.

## 2.2. Avaliação da Função Objetivo

A função apresentada pela equação (1a), e a partir de agora denominada  $f(x)$ , diz respeito às perdas de potência ativa com o sistema atuando de acordo com as variáveis de controle definidas por um vetor  $x$ . O objetivo do modelo é minimizar  $f(x)$ , que também pode ser chamada de função *fitness* por representar a aptidão de uma solução em minimizar as perdas ativas do SEE teste.

Entretanto, o FPOR possui restrições de desigualdade que devem ser levadas em consideração a fim de concluir sobre a qualidade de uma solução. Neste trabalho, a violação de uma restrição é adicionada à função objetivo através de uma penalização, evitando assim, a necessidade de se trabalhar com a otimização de um sistema multiobjetivo. A função penalizada  $g(x)$  apresentada na equação (2) recebe o nome de função *unfitness* e representa o nível de infactibilidade de uma solução em relação às restrições do sistema. Onde  $\rho$  é um número consideravelmente elevado e representa a penalização a ser aplicada.

$$g(x) = \rho \left( \sum_{k \in \mathcal{B}_c} \Delta V_k + \sum_{i,j \in \mathcal{L}} \Delta S_{ij} + \sum_{m \in \mathcal{B}_g} \Delta Q_{g_m} \right) \quad (2)$$

A função *unfitness* é determinada pelas violações das restrições de desigualdade relativas à tensão nas barras (em pu), do fluxo de potência nas linhas de transmissão (em MVA) e da geração de potência reativa nas barras de geração (em MVAR), as quais são calculadas de acordo com as equações (3), (4) e (5), respectivamente.

$$\Delta V_k = \begin{cases} V_k^{min} - V_k, & \text{se } V_k < V_k^{min} \\ V_k - V_k^{max}, & \text{se } V_k > V_k^{max} \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

$$\Delta S_{ij} = \begin{cases} -S_{ij} - S_{ij}^{max}, & \text{se } S_{ij} < -S_{ij}^{max} \\ S_{ij} - S_{ij}^{max}, & \text{se } S_{ij} > S_{ij}^{max} \\ 0, & \text{caso contrário} \end{cases} \quad (4)$$

$$\Delta Q_{g_m} = \begin{cases} Q_{g_m}^{min} - Q_{g_m}, & \text{se } Q_{g_m} < Q_{g_m}^{min} \\ Q_{g_m} - Q_{g_m}^{max}, & \text{se } Q_{g_m} > Q_{g_m}^{max} \\ 0, & \text{caso contrário} \end{cases} \quad (5)$$

Por fim, a função objetivo  $h(x)$  a ser minimizada encontra-se expressa na equação (6). Ela é composta pelas funções *fitness* e *unfitness* definidas anteriormente.

$$h(x) = f(x) + g(x) \quad (6)$$

## 3. Métodos de Otimização

Para realizar o ajuste das variáveis de controle do modelo FPOR são apresentados nesta seção três métodos de otimização bio-inspirados a serem utilizados: o Algoritmo Genético, o Algoritmo Genético de Chu e Beasley e o algoritmo Otimização por Enxame de Partícula, do inglês *Particle Swarm Optimization* (PSO).

### 3.1. Algoritmo Genético

Os AGs são métodos de otimização propostos inicialmente por Holland [1992]. Baseados no processo de evolução das espécies proposta por Charles Darwin, algoritmos desta classe possuem etapas que simulam a biologia evolutiva como a seleção, recombinação e mutação, promovendo assim, a manutenção dos indivíduos melhor adaptados. A Figura 1 apresenta o pseudocódigo de um Algoritmo Genético.

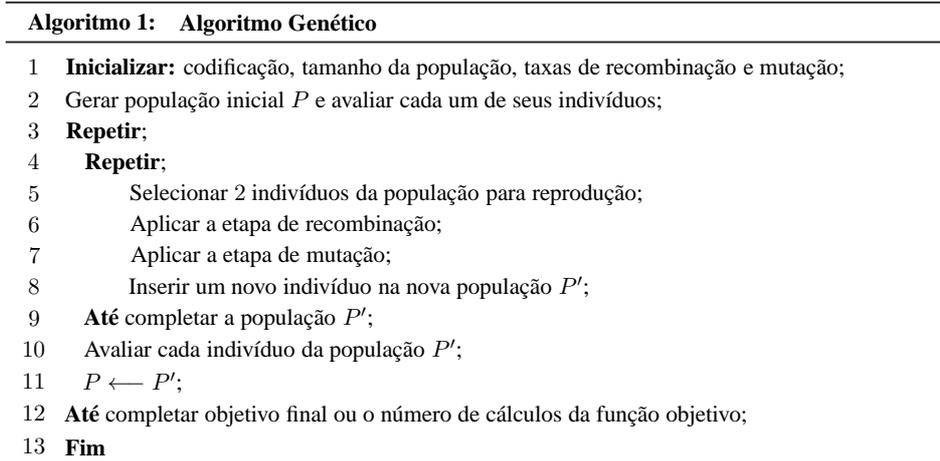


Figura 1: Pseudocódigo de um Algoritmo Genético.

Na sua forma básica, os AGs geram a população inicial de forma aleatória. Na sequência, as soluções são avaliadas e ocorre uma etapa de seleção, na qual os indivíduos melhor adaptados, de acordo com a função objetiva apresentada na equação (6), recebem o direito de se reproduzir. A reprodução é simulada pelas etapas de recombinação e mutação. Por fim, o processo é reiniciado com a substituição integral da população corrente ( $P'$ ) na população inicial ( $P$ ). O processo é repetido até que a condição de parada seja satisfeita.

#### 3.1.1. Seleção

Neste trabalho, a etapa de seleção é realizada por torneio Goldberg [1989]. Dessa forma, as soluções da população corrente são comparadas em jogos sucessivos mantendo-se as vencedoras até que restem apenas duas soluções candidatas à reprodução. A Figura 2 ilustra um torneio para seleção de dois indivíduos em uma determinada população  $P$ .

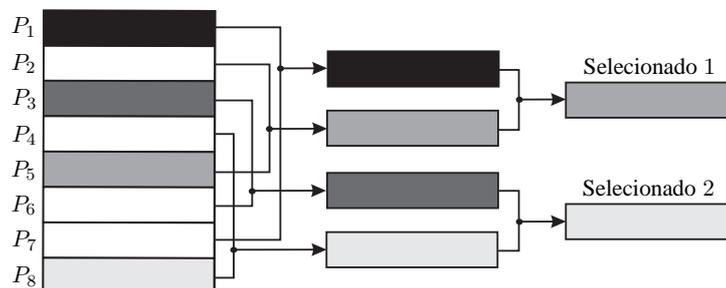


Figura 2: Seleção por Torneio.

#### 3.1.2. Recombinação

A etapa de recombinação ou *crossover* é realizada por um operador de cruzamento multipontos Goldberg [1989]. A quantidade de pontos de cruzamento é diretamente proporcional à dimensão do problema e a partir disto são definidas  $n$  regiões de corte para troca de genes. Os pontos de corte são definidos de forma aleatória, desde que não haja a sobreposição em relação à outra região de corte. A Figura 3 apresenta a recombinação entre dois indivíduos anteriormente selecionados.

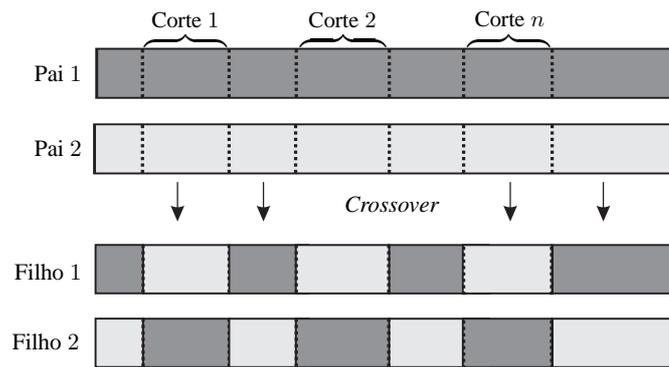


Figura 3: Recombinação por Cruzamento Multipontos.

### 3.1.3. Mutação

A etapa de mutação finaliza o processo de reprodução, para tal é considerada uma taxa de mutação onde todos os genes dos descendentes gerados possuem a mesma probabilidade de serem escolhidos. Para cada gene  $x_k$  escolhido para mutação, uma pequena variação ou perturbação é aplicada a ele conforme demonstra a equação (7).

$$x'_k = x_k \pm \Delta x_k^t \quad (7)$$

Na equação (7),  $\Delta x_k^t$  é a variação que será aplicada ao gene  $x_k$  de acordo com o tipo  $t$  da variável que ele contém. A variação pode ser negativa ou positiva com igual probabilidade de ocorrência. Para as variáveis discretas, a variação é o próprio passo de discretização. Quando a variável é binária, o gene mutacionado  $x'_k$  passa de 0 para 1 ou de 1 para 0. Com relação às variáveis contínuas, a perturbação é dimensionada proporcionalmente aos limites que as variáveis podem assumir. A Figura 4 apresenta uma interpretação visual do processo de mutação implementado para as variáveis contínuas e discretas.

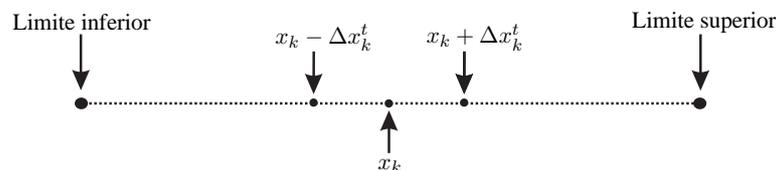


Figura 4: Mutação de um Gene nos Algoritmos Genéticos.

Ao final da etapa de mutação são gerados descendentes que se colocam como candidatos a compor a nova população. No AG utilizado, os descendentes gerados são necessariamente substituídos na nova população do ciclo geracional.

## 3.2. Algoritmo Genético de Chu e Beasley

As alterações propostas pelo AGCB em relação à versão básica dos AGs incluem uma codificação própria para cada tipo de problema, uma etapa de melhoria local, a avaliação das funções *fitness* e *unfitness* separadamente e a manutenção da diversidade na população assim como a preservação de soluções de ótima qualidade [Chu e Beasley, 1997].

### 3.2.1. Seleção, Recombinação e Mutação

As etapas de seleção e recombinação para o AGCB sofrem pequenas alterações em relação às etapas apresentadas para o AG. Para o AGCB utilizado neste trabalho, a etapa de seleção considera a função *fitness* apresentada na equação (1a) durante a realização do torneio. Em relação à recombinação, a diferença está na preservação de apenas um dos descendentes gerados a partir do cruzamento multipontos. A etapa de mutação do AGCB não sofre alterações em relação ao AG apresentado. A Figura 5 apresenta o pseudocódigo de um AGCB.

---

**Algoritmo 2: Algoritmo Genético de Chu e Beasley**

---

- 1 **Inicializar:** codificação, tamanho da população, taxas de recombinação e mutação;
  - 2 Gerar população inicial e torná-la a população corrente;
  - 3 Encontrar o *fitness* e *unfitness* para todos os indivíduos;
  - 4 **Repetir;**
  - 5     Selecionar 2 indivíduos da população para reprodução;
  - 6     Aplicar a etapa de recombinação e preservar apenas um descendente;
  - 7     Aplicar a etapa de mutação;
  - 8     Aplicar a etapa de melhoria local;
  - 9     Avaliar se o descendente deve entrar na população corrente;
  - 10 **Até** completar objetivo final ou o número de cálculos da função objetivo;
  - 11 **Fim**
- 

Figura 5: Pseudocódigo de um Algoritmo Genético de Chu e Beasley.

### 3.2.2. Melhoria Local

Uma das propostas do AGCB é a etapa de melhoria local, a intenção é realizar uma busca local a fim de elevar a qualidade de um descendente  $x$  gerado pelas etapas de recombinação e mutação. Neste trabalho é aplicada uma busca local simples baseada no conceito de sensibilidade.

Inicialmente, calcula-se a função objetivo  $h(x)$  do indivíduo  $x$  a ser melhorado conforme equação (6). Na sequência, seleciona-se aleatoriamente um gene  $x_k$  deste indivíduo e aplica-se a variação apresentada pela equação (7).

Se a função objetivo do novo indivíduo  $h(x')$  for melhor (menor) que a função objetivo do indivíduo corrente  $h(x)$ ,  $x'$  se torna o indivíduo corrente e mantendo-se o sinal da equação (7), são aplicadas variações sucessivas a ele até que o número de iterações  $i_{bl}$  seja satisfeito ou que a função objetivo do indivíduo gerado na iteração atual  $h(x')$  não seja melhor que a função objetivo do indivíduo corrente  $h(x)$ .

Caso contrário, o sinal na equação (7) é trocado e o processo se repete até que o mesmo critério de parada descrito anteriormente seja atendido. O processo pode ser encerrado logo nas primeiras iterações, caso não seja possível melhorá-lo após a aplicação de variações positivas e negativas. A busca local é repetida para  $n_s$  genes, sendo  $n_s$  e  $i_{bl}$  números pequenos a fim de não exigir esforços computacionais elevados. Esse processo é aplicado para variáveis contínuas e discretas, quanto às variáveis binárias a mudança é de 0 para 1 e vice-versa, retornando para o valor original caso não haja melhora no indivíduo.

### 3.2.3. Substituição do Descendente na População

Por fim, o AGCB avalia se o descendente melhorado deve integrar a população a partir da análise de seus valores *fitness* da equação (1a) e *unfitness* da equação (2). A intenção desta etapa é garantir a diversidade e a qualidade das soluções da população corrente. No primeiro momento é verificado se o descendente é igual a qualquer um dos indivíduos integrantes da população, sendo descartado em caso positivo.

A base de regras inicia avaliando se o descendente é infactível, ou seja, se ele possui *unfitness* diferente de zero. Caso possua, busca-se na população corrente um indivíduo com maior infactibilidade em relação ao descendente. Caso seja encontrado, o descendente substitui este indivíduo. Caso contrário, o descendente é descartado.

Se o descendente é factível, ou seja, se ele possui *unfitness* igual a zero, verifica-se a existência de algum indivíduo infactível na população corrente. Caso exista, o descendente substitui o indivíduo mais infactível da população corrente. Caso todos os indivíduos da população corrente sejam factíveis, busca-se na população a existência de um indivíduo que tenha um valor *fitness* de pior qualidade em relação ao descendente. Se encontrado, o descendente substitui o indivíduo encontrado. Caso contrário, o descendente é descartado.

### 3.3. Algoritmo Particle Swarm Optimization

O algoritmo de otimização por enxame de partículas baseia-se no comportamento de pássaros durante a revoada na busca por alimentos. Observou-se que os pássaros alteravam suas posições de maneira coordenada, onde cada indivíduo ou partícula obedecia a três conceitos básicos: harmonia, separação e alinhamento. Com estes conceitos, os indivíduos evitavam colisões, não ocupavam ou se aproximavam de posições já ocupadas por outras partículas e ainda de maneira geral seguiam o fluxo do enxame [Rao, 2009].

Cada partícula se movimenta a partir de sua velocidade dada pela equação (8), a qual é atualizada a cada iteração de acordo com: um fator de busca  $W$ , as variáveis de ponderação  $c_1$  e  $c_2$ , número aleatórios  $r_1$  e  $r_2$  entre 0 e 1, além dos fatores cognitivos  $co_i$  e sociais  $so_i$  dados pelas equações (10) e (11), respectivamente. Partindo da posição em uma determinada transição  $t$ , a posição da partícula na próxima transição é dada em (9).

$$v_i(t+1) = Wv_i(t) + c_1r_1co_i(t) + c_2r_2so_i(t) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

$$co_i(t) = Pbest_i - x_i(t) \quad (10)$$

$$so_i(t) = Gbest - x_i(t) \quad (11)$$

Nas equações (8)–(11),  $Pbest_i$  é a melhor posição ocupada pela partícula  $i$ ,  $Gbest$  é a melhor posição ocupada entre todas as partículas. Para as variáveis discretas de uma partícula, a posição atual calculada é arredondada para o passo de discretização imediatamente superior respeitando-se os limites da variável em questão, caso não seja possível, a variável então é arredondada para o passo de discretização inferior mais próximo. Quanto às variáveis binárias é realizada uma ponderação de acordo com a equação (12).

$$x_i(t+1) = \begin{cases} 1 & \text{se } r_3 < S(v_i(t+1)) \\ 0 & \text{se } r_3 \geq S(v_i(t+1)) \end{cases} \quad (12)$$

Na equação (12),  $r_3$  é um número aleatório (entre 0 e 1) gerado a cada ponderação.  $S(.)$  é a função sigmoide com taxa de inclinação unitária e centrada na origem. A Figura 6 apresenta o pseudocódigo do algoritmo PSO.

<b>Algoritmo 3:</b> Particle Swarm Optimization	
1	<b>Inicializar:</b> tamanho da população, $W$ , $c_1$ , $c_2$ , $v_{max}$ , $x_{max}$ , $x_{min}$ ;
2	Determinar posições e velocidades iniciais aleatoriamente: $x_i(0)$ , $v_i(0)$ ;
3	Calcular e avaliar a função objetivo para todas as partículas: $f_o(x_i)$ ;
4	<b>Se</b> (função objetivo é melhor): Atualize $Gbest$ e $Pbest_i$ ;
5	<b>Se não:</b> Atualize $Pbest_i$ ;
6	<b>Enquanto</b> (condição de parada não for satisfeita):
7	<b>Para</b> (cada partícula) <b>faça:</b>
8	Calcular fatores cognitivos e sociais: $co_i$ e $so_i$ ;
9	Faça: $t \leftarrow t + 1$ ;
10	Atualizar velocidade: $v_i$ ;
11	Atualizar posição: $x_i$ ;
12	Calcular e avaliar a função objetivo: $f_o(x_i)$ ;
13	<b>Se</b> (função objetivo é a melhor encontrada pela partícula): Atualize $Pbest_i$ ;
14	<b>Se</b> (função objetivo é a melhor encontrada pelo algoritmo): Atualize $Gbest$ ;
15	<b>Fim</b>

Figura 6: Pseudocódigo do algoritmo PSO.

Durante o processo é possível que as variáveis contínuas e discretas das partículas busquem ocupar posições fora da região de busca. Portanto, são definidos limites inferiores ( $x_{min}$ ) e superiores ( $x_{max}$ ) para as variáveis dos indivíduos. A velocidade máxima de transição ( $v_{max}$ ) também é definida. Os limites a serem verificados a cada transição estão expressos nas equações (13) e (14).

$$v_i(t+1) = \begin{cases} v_i(t+1) & \text{se } -v_{max} \leq v_i(t+1) \leq v_{max} \\ -v_{max} & \text{se } v_i(t+1) < -v_{max} \\ v_{max} & \text{se } v_i(t+1) > v_{max} \end{cases} \quad (13)$$

$$x_i(t+1) = \begin{cases} x_i(t) + v_i(t+1) & \text{se } \geq x_{min} \text{ e } \leq x_{max} \\ x_{max} & \text{se } > x_{max} \\ x_{min} & \text{se } < x_{min} \end{cases} \quad (14)$$

#### 4. Codificação, Restrições e Definição dos Parâmetros dos Métodos

A codificação considera a existência de três tipos de variáveis diferentes: contínuas para as tensões nas  $k$  barras de geração, discretas para os taps dos  $i$  transformadores em fase e binárias para representar o estado dos  $n$  bancos de capacitores e reatores *shunts*. A Figura 7 apresenta a codificação a ser utilizada.

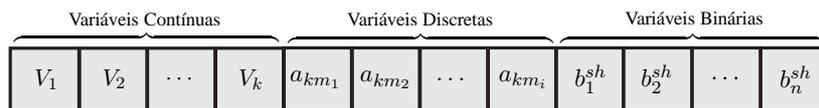


Figura 7: Representação de um indivíduo para os métodos de otimização.

Neste trabalho, as variáveis de controle dos sistemas testes utilizados para as simulações podem assumir os seguintes valores: as tensões nas barras devem estar entre 0,94 e 1,06 pu, os taps dos transformadores assumem valores discretos entre 0,9 e 1,1 pu, os bancos de capacitores e reatores *shunt* assumem 0 se estiverem desligados e 1 se estiverem ligados. Em todos os métodos utilizados, no decorrer de qualquer processo, é aplicada uma técnica para garantir que toda vez que uma variável tente assumir um valor fora dos limites estabelecidos ela seja impedida e seu valor se torne exatamente o limite estabelecido.

O passo de discretização das variáveis discretas referentes aos transformadores é de 0,01. Para utilização da equação (7), também é definido um passo  $\Delta x_k^t$  para as variáveis contínuas que vale 0,005. As variáveis binárias apenas alternam entre 0 e 1.

As taxas de recombinação e mutação utilizadas variam a cada ciclo geracional entre: 25% a 60% e 1% a 5%. Os parâmetros do AGCB são definidos em:  $n_s = 3$  e  $i_{bl} = 5$ . O tamanho da população para os métodos aplicados é de 32 indivíduos. Os parâmetros do PSO são definidos em:  $W = 1$ ;  $c_1 = c_2 = 2,0$ ;  $v_{max}$  é definido como 10% de  $x_{max}$  que é o limite superior da variável associada. O fator penalização  $\rho$  das infactibilidades vale  $10^7$ .

#### 5. Simulações e Resultados

Os métodos de otimização apresentados foram implementados com o auxílio do software Matlab<sup>®</sup> em um computador que dispõe de 12 GB de memória RAM. Os cálculos do fluxo de potência são realizados pelo método de Newton-Raphson através do software MatPower<sup>®</sup> [Zimmerman et al., 2011]. São realizados 100 testes para cada meta-heurística considerando os sistemas elétricos IEEE 57, 118 e 300 [Power Systems Test Case Archive, 1999], onde o teste para cada sistema considera, respectivamente, 20.000, 40.000 e 60.000 cálculos da função objetivo como critério de parada. A Tabela 1 apresenta os tempos médios de processamento.

Ao analisar a Tabela 1, verifica-se que ainda que não existam grandes diferenças entre os tempos de processamento dos métodos, o AG e o PSO são os que possuem, em média, os menores tempos. Entretanto, este fator não é preponderante para a análise da qualidade dos métodos, pois

Tabela 1: Desempenho dos Métodos de Otimização.

Sistema	AG	AGCB	PSO
	$\Delta t(s)$	$\Delta t(s)$	$\Delta t(s)$
IEEE 57	58,21	62,89	59,27
IEEE 118	178,07	189,12	180,23
IEEE 300	562,52	593,94	565,81

este trabalho não tem como objetivo realizar os ajustes das variáveis em tempo real. O fator preponderante na comparação da qualidade dos métodos é sua capacidade de fornecer ajustes das variáveis de controle dos sistemas que garantam a minimização das perdas ativas atendendo às restrições físicas e operacionais. Em vista disto, a Tabela 2 apresenta as soluções obtidas pelos métodos a partir das simulações realizadas nos sistemas elétricos testes.

Tabela 2: Soluções Obtidas pelos Métodos de Otimização.

Sistema	Soluções	AG	AGCB	PSO
		Perdas (MW)	Perdas (MW)	Perdas (MW)
IEEE 57	Melhor	25,3328	24,6918	25,0141
	Média	26,2103	25,5386	25,8108
	Pior	26,9782	26,6548	26,8978
IEEE 118	Melhor	122,8547	117,8521	118,8241
	Média	125,9873	119,4874	121,2576
	Pior	129,9471	122,4572	125,7956
IEEE 300	Melhor	394,8785	381,7589	384,1254
	Média	408,5417	388,7624	392,1345
	Pior	420,4875	408,5984	413,6578

Observa-se que o AG apresenta os piores resultados dentre os métodos utilizados, isso se deve à dificuldade do algoritmo em sair de ótimos locais, haja vista a perda da diversidade durante o processo. Apesar do PSO apresentar soluções de ótima qualidade, as menores perdas de potência ativa são obtidas pelo AGCB, tal fato se explica pela forma com que as infactibilidades são tratadas neste algoritmo. No primeiro momento, o AGCB procura diminuir as violações de restrições a fim de minimizar as infactibilidades até que tendam a zero. Além disso, a estratégia de substituição de indivíduos na população garante a diversidade e qualidade das soluções, promovendo assim, melhoras sucessivas nas perdas ativas do sistema. As Figuras 8 a 10 ilustram a dinâmica de eliminação das infactibilidades e minimização das perdas ativas pelo AGCB para os sistemas testados. Para tal, escolheu-se a melhor solução obtida por este método de otimização.

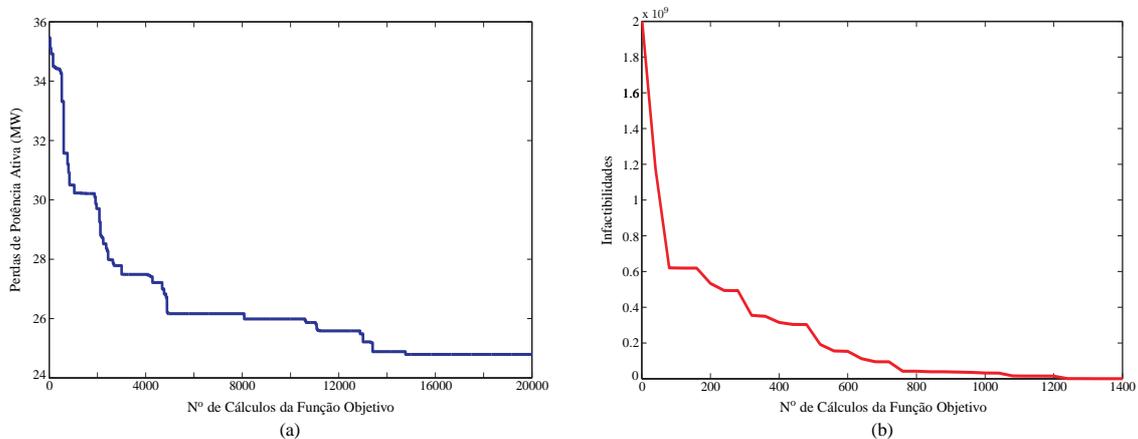


Figura 8: Perdas ativas (a) e infactibilidades (b) para o sistema IEEE 57 otimizado pelo AGCB.

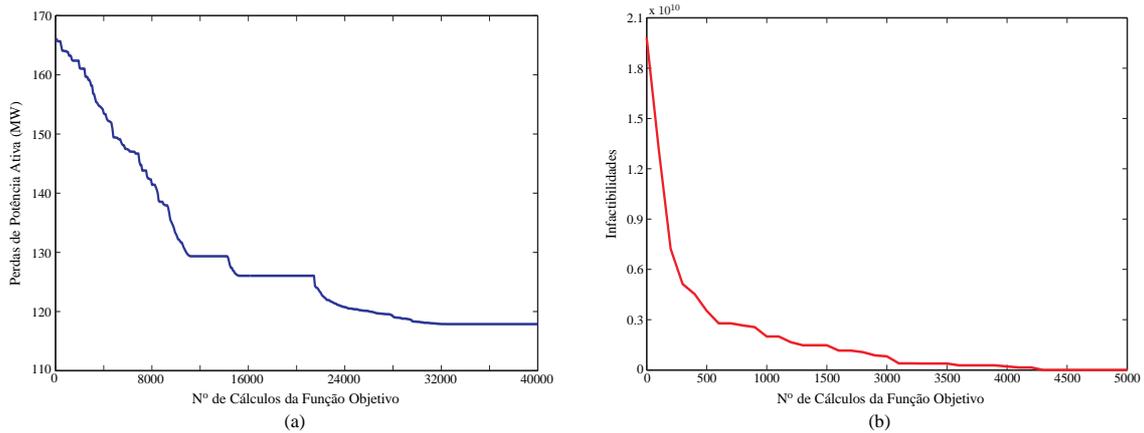


Figura 9: Perdas ativas (a) e infactibilidades (b) para o sistema IEEE 118 otimizado pelo AGCB.

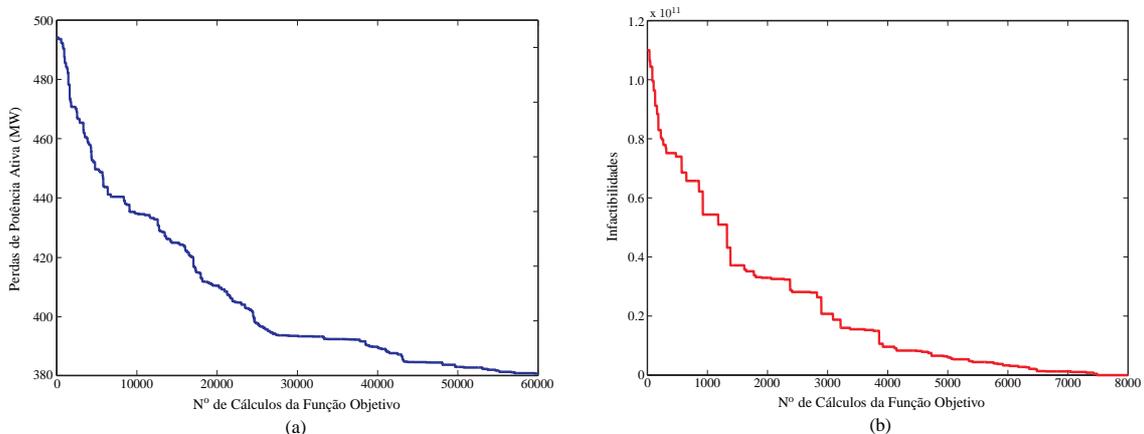


Figura 10: Perdas ativas (a) e infactibilidades (b) para o sistema IEEE 300 otimizado pelo AGCB.

## 6. Conclusões

Neste trabalho foram utilizados três métodos de otimização (AG, AGCB e PSO) no ajuste das variáveis de controle de um modelo FPOR com o intuito de minimizar as perdas ativas dos sistemas testes IEEE 57, 118 e 300.

Os resultados obtidos mostram que dentre os métodos utilizados, o AGCB é a técnica melhor adaptada a este tipo de problema, já que possui uma forma adequada de tratamento das infactibilidades e estratégias que garantem a diversidade e qualidade das soluções, evitando a estagnação em ótimos locais. Portanto, este trabalho evidencia e reforça a qualidade do AGCB como uma ferramenta na resolução do problema de FPO.

## 7. Agradecimentos

Este trabalho teve o apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

## Referências

- Abido, M. (2002). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, 24(7):563 – 571.
- Bakirtzis, A. G., Biskas, P. N., Zoumas, C. E., e Petridis, V. (2002). Optimal power flow by enhanced genetic algorithm. *IEEE Transactions on Power Systems*, 17(2):229–236.
- Biskas, P. N., Ziogos, N. P., Tellidou, A., Zoumas, C. E., Bakirtzis, A. G., Petridis, V., e Tsakoumis, A. (2005). Comparison of two metaheuristics with mathematical programming methods for the

- solution of opf. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, p. 6 pp.
- Boucekara, H., Abido, M., e Boucherma, M. (2014). Optimal power flow using teaching-learning-based optimization technique. *Electric Power Systems Research*, 114:49–59.
- Carpentier, J. L. (1962). Contribution à l'étude du dispatching économique. *Bulletin de la Société Française des Electriciens*, 3(8):431–447.
- Chu, P. e Beasley, J. (1997). A genetic algorithm for the generalised assignment problem. *Computers & Operations Research*, 24(1):17–23.
- Dommel, H. W. e Tinney, W. F. (1968). Optimal power flow solutions. *IEEE Transactions on Power Apparatus and Systems*, PAS-87(10):1866–1876.
- Glover, F. e Kochenberger, G. (2003). *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*. Springer US.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Artificial Intelligence. Addison-Wesley Publishing Company.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Bradford book. MIT Press.
- Jabr, R. A., Coonick, A. H., e Cory, B. J. (2002). A primal-dual interior point method for optimal power flow dispatching. *IEEE Transactions on Power Systems*, 17(3):654–662.
- Kennedy, J. e Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, p. 1942–1948.
- Papalexopoulos, A. D., Imparato, C. F., e Wu, F. F. (1989). Large-scale optimal power flow: Effects of initialization, decoupling discretization. *IEEE Power Engineering Review*, 9(5):72–74.
- Power Systems Test Case Archive, U. (1999). Disponibiliza sistemas testes IEEE. Disponível em: <<http://www.ee.washington.edu/research/pstca/>>. Acessado: 2016-03-15.
- Quintana, V. H., Torres, G. L., e Medina-Palomo, J. (2000). Interior-point methods and their applications to power systems: a classification of publications and software codes. *IEEE Transactions on Power Systems*, 15(1):170–176.
- Ramos, J. L. M., Exposito, A. G., e Quintana, V. H. (2005). Transmission power loss reduction by interior-point methods: implementation issues and practical experience. *IEE Proceedings - Generation, Transmission and Distribution*, 152(1):90–98.
- Rao, S. S. (2009). *Engineering optimization: theory and practice*. New Jersey.
- Sousa, A. A., Torres, G. L., e Canizares, C. A. (2011). Robust optimal power flow solution using trust region and interior-point methods. *IEEE Transactions on Power Systems*, 26(2):487–499.
- Zhu, J. (2009). *Optimization of power system operation*. IEEE Press series on power engineering. Wiley-IEEE, Piscataway NJ.
- Zimmerman, R. D., Murillo-Sanchez, C. E., e Thomas, R. J. (2011). Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19.